ADA047983

ARPA ORDER NO.: 189-1
7D20 Cybernetics Technology

R-1940/1-ARPA
October 1977

A Dynamic Model for Optimum Bonus Management: Computer Program and Mathematical Analysis

Roy Danchick

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

The research described in this report was sponsored by the Defense Advanced Research Projects Agency under Contract No. DAHC15-73-C-0181.

Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14 RAND/

1. REPORT NUMBER: R-1940/1-ARPA
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

Optimum

4. TITLE (and Subtitle): 6 A Dynamic Model for ~~Optimal~~ Bonus Management: Computer Program and Mathematical Analysis,
5. TYPE OF REPORT & PERIOD COVERED: 9 Interim rept.
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): 10 Ray/Danchick
8. CONTRACT OR GRANT NUMBER(s): 15 DAHC15-73-C-0181, ARPA Order-189
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The Rand Corporation, 1700 Main Street, Santa Monica, Ca. 90406
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, Department of Defense, Arlington, Va. 22209
12. REPORT DATE: 11 October 1977
13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 12 65p.
15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Military Personnel
Military Specialties
Military Bonuses
Manpower
Requirements
Personnel Management
Computer Programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

296 600

1B

This report

Describes the mathematics and the computer program for solving the problem of optimum bonus management formulated in Patricia Munch, 'A Dynamic Model for Optimum Bonus Management', ~~R-1940-ARPA, January 1977~~. The problem of optimum bonus management is treated as a discrete linear control system with a quadratic cost function and solved by using Pontryagin's discrete maximum principle. The state of the system at discrete time is a vector of numbers of men in each of a set of year groups. The system evolves linearly in time under linear controls that are the bonuses paid to the men in a prescribed subset of the year groups. The program solves for the sequence of bonus values that drive a given inital state of year groups to a prescribed final state and minimize a sum of quadratic bonus and penalty costs. The program, which consists of a main program and 22 subroutines, is written in FORTRAN IV double precision for the IBM 370-158. (Author)

ARPA ORDER NO.: 189-1
7D20 Cybernetics Technology

R-1940/1-ARPA
October 1977

A Dynamic Model for Optimum Bonus Management: Computer Program and Mathematical Analysis

Roy Danchick

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

PREFACE

This report was prepared under Rand's DoD Training and Manpower Program, sponsored by the Cybernetics Technology Office of the Defense Advanced Research Projects Agency (ARPA). The purpose of this research program is to develop broad strategies and specific solutions for dealing with present and future military manpower problems, including the development of new research methodologies for examining broad classes of manpower problems, as well as specific problem-oriented research. It is hoped that this research program will provide analysis of current and future manpower issues and contribute to a better understanding of the manpower problems confronting the Department of Defense.

This report is a companion to Patricia Munch, *A Dynamic Model for Optimum Bonus Management*, R-1940-DoD/ARPA, January 1977, which deals with the institutional background of reenlistment bonus management and its formulation both as an economic problem and as a mathematical problem of optimal control.

The present report describes the mathematical foundations, operating instructions, and structure and function of the Dynamic Bonus Management Model Program (DBMMP) and its subroutines. Its purpose is to give a clear, detailed derivation of the formulas used by the program to determine the optimal bonus policy. It provides complete, clear DBMMP input parameter definitions and input parameter data card format specifications that will permit the user to run the program with ease. The report also supplies the programmer with sufficient knowledge of program function, architecture, and logical flow to modify the program, as required, for greater generality.

SUMMARY

This report provides a detailed mathematical derivation of the algorithm for solving the optimal bonus management problem that was posed in Patricia Munch, *A Dynamic Model for Optimum Bonus Management*, R-1940-DoD/ARPA, January 1977. In addition to the mathematical development, the report also describes the computer program that incorporates the solution method and defines the economic setting of the problem.

The problem of optimum bonus management is treated as a discrete linear control system in which the optimal control is determined from Pontryagin's discrete maximum principle.

In this formulation the state of the system at time t is a vector, each of whose components is the number of men in each of a number of year groups. The system evolves linearly in discrete time steps under the linear action of a set of control variables. The control variables are bonuses that are paid to the men in certain year groups.

The optimization problem involves the determination of the sequence of bonuses that will drive prescribed initial state of year group distribution to a desired terminal state in a given number of years while minimizing a cost function. The cost function is a cumulative sum of two quadratic forms, penalty cost and bonus cost. The penalty cost quadratic form represents the costs accruing to shortages or overages in the year group numbers relative to the corresponding desired terminal state values.

Both penalty and cost quadratic term components are functions of a set of economic model parameters that are defined in the introductory section. The introduction also indicates which parameter values are program constants and which must be supplied by the program reader. The functional relation between the economic model parameters and the components of the quadratic forms are developed in App. B.

The computer program incorporating the solution method is written in FORTRAN IV double precision for the IBM 370-158. It contains a main program and 22 subroutines.

Although the description in the sequel is oriented toward the solution of the general bonus management optimal control problem, the program is essentially a "one-shot" research task and has been "hard coded" for a system of eight year groups with bonuses for men going into the first and fifth year groups. This limitation should be removable without a major reprogramming effort.

CONTENTS

TABLES

I. INTRODUCTION

This report describes the mathematical solution method of the computer program that puts into operation the model described in F. P. Munch, *A Dynamic Model for Optimal Bonus Management*, The Rand Corporation, R-1940-DoD/ARPA, January 1977. The purpose of this modeling effort is to structure the problem of bonus management as an economic problem, incorporating into a single decisionmaking framework the several criteria currently used in setting bonuses. The dynamic adjustment model determines the optimum level of bonuses for several year groups in a military specialty, at successive points in the planning horizon, subject to the constraint that the deviations of the actual from the desired manpower inventory be eliminated over the assigned period. The optimum structure of bonuses minimizes the sum of two sources of cost--bonus cost incurred to reduce shortages and penalty cost assigned to shortages and overages. Penalty cost is a quadratic form in the deviations of actual from desired manning levels, where the quadratic coefficients are derived from the production and demand functions for specialty output. These functions aggregate men of differing skill levels into a homogeneous unit of "output" (man-years) and assign a dollar value to this output. Bonus cost is derived from the supply function, which relates number of men to the level of bonus paid. Those concepts currently used as rules of thumb by bonus managers, such as criticality of a specialty, substitutability between men in different year groups, and responsiveness of supply to a bonus, are characterized by parameters in the component functions of the model. The values of these parameters are to be supplied by the user. (See Table 1.) The component functions of the model are restated below to facilitate specification of the values of the input parameters by the user. Section II contains a statement of the mathematical problem and its solution.

Table 1

DEFINITION OF VARIABLES

Variable	Definition	To Be Supplied by User
Z_t	= specialty "output" in year t, a weighted sum of number of men in each term of service.[a]	
$X_{i,t}$	= number of man years in ith term of service in year t.[b] $= \sum_j L^i_{j,t}$.	
$X_{i,T}$	= desired number of man years in ith term of service.	✓
A_j	= no-bonus supply of lateral entrants to the specialty. In the current model only A_1 is non-zero.	✓
$L^i_{j,t}$	= number of men in the jth year group (YOS) in ith term of service in year t.	
$L^i_{j,T}$	= desired number of men in the jth year group (YOS) in year T.	✓
δ_i	= distribution parameter, reflecting relative productivity of men in different terms of service. To select appropriate values, note that $\frac{\text{Marginal product } X_1}{\text{Marginal product } X_2} = \frac{\delta_1}{\delta_2}\left(\frac{X_1}{X_2}\right)^{1+\rho}$, and $\sum_1 \delta_i = 1$.	✓
ρ	= substitution parameter, defined as follows: $\sigma = 1/(1 + \rho)$, where σ = elasticity of substitution between X_1 and X_2. Since $0 \leq \sigma \leq + \infty$, admissible values of ρ run from -1 to + ∞.	✓
μ	= returns to scale parameter $\mu \gtrless 1$ implies an equiproportionate increase in all inputs by x% leads to an increase in output $\gtrless$ x%.	✓

Table 1--Continued

Variable	Definition	To Be Supplied by User
α_j	= no-bonus continuation rate, $j = 1,2 \ldots 7$--i.e., percentage of men in YOS that continue into YOS_{j+1} if no bonus is paid.	✓
B_{jt}	= bonus paid to YOS_j in year t.	
β_j	= slope of supply function.	✓
P	= price per unit of specialty output. If the dynamic adjustment model is run in conjunction with a steady-state model, P is set equal to the value of the Lagrange multiplier obtained by solving for the cost-minimizing input mix, subject to producing the target output level, Z_T. If the dynamic model is run without a steady-state model, P may be assigned by setting the target inventory and one wage level, say W_2, at arbitrary but reasonable levels and solving from the first order condition of a steady-state model: $W_2 = P\delta_2 \left(\frac{Z_T}{L_{2T}}\right)^{1+\rho}$.	✓
$1/\varepsilon$	= elasticity of price with respect to quantity. If a deviation of specialty output from the target level of x% results in a change of more (less) than x%, ε should be set less (greater) than unity. ε reflects the criticality of a specialty.	✓
δ	= a scale parameter.[c]	

[a]For a discussion and economic interpretation of this production function see K. Sato, "A Two-Level Constant-Elasticity-of-Substitution Production Function," *Review of Economic Studies*, Vol. 34, April 1967.

[b]For an alternative specification, allowing differences in productivity of men in different year groups within a term of service, see F. P. Munch, *A Dynamic Model for Optimal Bonus Management*, R-1940-DoD/ARPA, January 1977.

[c]δ is denoted α in Munch (1977). The notation is changed here to avoid confusion with the continuation rate parameters.

PRODUCTION FUNCTION[1]

$$Z_t = \left[\sum_{i=1}^{2} \delta_i X_{i,t}^{-\rho} \right]^{-\frac{\mu}{\rho}} .$$

SUPPLY FUNCTION

$$L_{j,t} = A_j + \alpha_{j-1} L_{j-1,t-1} + \beta_j B_{j,t} ,$$

where $A_j = 0,\ j > 1$.

DEMAND FUNCTION

$$P_t = \delta Z_t^{-(1/\varepsilon)} .$$

---

[1]The functions are stated in the eight-year group, two-bonus version of the model currently hard-coded. For a more general statement, see Munch (1977).

II. THE OPTIMAL CONTROL PROBLEM

DEFINITION OF SYMBOLS

$$L_t' = [L_{2,t}, L_{3,t}, \ldots, L_t]' \qquad (\text{' denoting transpose})$$

is the reduced state of the system in year t.

$$B_t' = [B_{1,t}, B_{2,t}, \ldots, B_{1+4[\frac{m-1}{4}],t}]' ,$$

where [x] is the largest integer ≤ x, is the vector of bonus control variables for year n.

$$A = \begin{bmatrix} 0 & & \cdots & . & 0 \\ \alpha_2 & 0 & \cdots & & 0 \\ 0 & \alpha_3 & & & 0 \\ \vdots & & & & \vdots \\ 0 & & & \alpha_{m-1} & 0 \end{bmatrix}_{m-1\times m-1}$$

is the reduced[1] state transition matrix. $\alpha_2, \alpha_3, \ldots, \alpha_{m-1}$ are no-bonus continuation rates.

---

[1]Although there are m year groups, the dimension of the system can be reduced by one because the first year group number, $L_{1,t}$ is a linear function of the control variable $B_{1,t}$:

$$L_{1,t} = K + \alpha_1 B_{1,t} .$$

$$\beta = \begin{bmatrix} \alpha_1\beta_1 & 0 & \cdots & 0 & \\ 0 & 0 & \cdots & 0 & \\ 0 & 0 & \cdots & 0 & \\ 0 & 0 & \cdots & 0 & \\ 0 & \alpha_5\beta_5 & \cdots & & \\ \vdots & & & & \\ 0 & & \cdots & \alpha_{1+4[\frac{m-1}{4}]} & \beta_{1+4[\frac{m-1}{4}]} \\ \vdots & & & \vdots & \\ 0 & & & 0 & \end{bmatrix}_{m-1\times 1+[\frac{m-1}{4}]}$$

is the control variables matrix multiplier.

$$C = \begin{bmatrix} \alpha_1 K_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{m-1\times 1}$$ (K_1 is constant no-bonus number of enlistees into the first YOS)

is the reduced constant first year group input.

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & \beta_5 & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & \beta_{1+4[\frac{m-1}{4}]} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}_{m-1\times[\frac{m-1}{4}]}$$

is the state-bonus modifier matrix.

$$F = \begin{bmatrix} F_{11} & F_{12} \\ & \\ F_{12} & F_{22} \end{bmatrix}_{m \times m}$$

is the negative of the positive semi-definite cost quadratic form. The matrix entries F_{ij} are functions of economic parameters and the desired terminal inventory. The economic significance of the F_{ij} are discussed in Munch (1977), pp. 18-19. Refer to App. B for the equations defining these components.

$$R = \text{diag}\left[\beta_1(1 - \beta_1 F_{11}),\ \beta_5,\ \beta_9,\ \ldots,\ \beta_{1+4[\frac{m-1}{4}]}\right]$$

is the bonus cost quadratic form.

F_2 = lower diagonal m-1 × m-1 submatrix of F is the penalty cost part of the cost quadratic associated with the reduced state L_t.

$$F_1 = \begin{bmatrix} F_{12} & & F_{13} & \cdots & F_{1,m} \\ 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}_{1+[\frac{m-1}{4}] \times m-1}$$

is the penalty cost matrix associated with $B_{1,t}$, the bonus for the first year group in year t. F_{ij}, j = 2, 3, ..., m are the last m-1 entries in the first row of F.

DEFINITION OF THE OPTIMAL CONTROL PROBLEM

The optimal control problem is to find a sequence of control vector (yearly bonuses) B_n such that

$$\sum_{t=0}^{T} f_n(L_t, B_t) \qquad \text{is a minimum subject to:}$$

$$L_{t+1} = AL_t + \beta B_t + C \ ,$$

$$L_T = \bar{L} \ , \qquad \text{a prescribed terminal inventory}$$

where

$$f_t = 1/2B_t'RB_t$$

$$-1/2(L_t + SB_t - \bar{L})' F_2(L_t + SB_t - \bar{L})$$

$$-h_1 B_t' F_1(L_t + SB_t - \bar{L}) \ .$$

SOLUTION TO THE OPTIMAL CONTROL PROBLEM

The solution to the optimal control problem is found by means of the discrete version of the maximum principle. Thus, if the system Hamiltonian, $H_t = H_t(L_t, \lambda_{t+1}, B_t)$, is defined as:

$$H_t = -f_t + \lambda'_{t+1}(AL_t + \beta B_t + C) \ ,$$

then the optimal solution must, at each stage t, satisfy

$$\frac{\partial H_t}{\partial B_t} = 0 \qquad (1)$$

$$\frac{\partial H_t}{\partial \lambda_{t+1}} = L_{t+1} \qquad (2)$$

$$\frac{\partial H_t}{\partial L_t} = \lambda_t \ . \qquad (3)$$

Equation (1) is seen to be equivalent to:

$$-RB_t + S'F_2(L_t + SB_t - \bar{L}) + \beta_1F_1(L_t - \bar{L})$$

$$+ \beta_1(F_1S + S'F_1') B_t + \beta'\lambda_{t+1} = 0$$

or

$$B_t = (R - S'F_2S - F_1^*)^{-1} [(S'F_2 + \beta_1F_1)(L_t - \bar{L}) + \beta'\lambda_{t+1}] , \qquad (4)$$

where

$$F_1^* = \beta_1(F_1S + S'F_1) .$$

Equation (2) will yield a re-statement of the transition equation constraint, while Eq. (3) yields:

$$\lambda_t = F_2(L_t + SB_t - \bar{L}) + \beta_1F_1'B_t + A'\lambda_{t+1} . \qquad (5)$$

Substituting the righthand side of Eq. (4) yields:

$$\lambda_t = F_2(L_t - \bar{L}) + A'\lambda_{t+1} \qquad (6)$$

$$+ (F_2S + \beta_1F_1')(R - S'F_2S - F_1^*)^{-1}[(S'F_2 + \beta_1F_1)(L_t - \bar{L}) + \beta'\lambda_{t+1}] .$$

Expanding, collecting terms, and solving for λ_{t+1} in terms of L_t and λ_t,

$$\lambda_{t+1} = B_1^{-1} [-A_1(L_t - \bar{L}) + \lambda_t] , \qquad (7)$$

where

$$A_1 = F_2 + (F_2S + \beta_1F_1')(R - S'F_2S - F_1^*)^{-1} (S'F_2 + \beta_1F_1) \qquad (8)$$

and

$$B_1 = [A' + (F_2S + B_1F_1')(R - S'F_2S - F_1^*)^{-1} \beta'] . \qquad (9)$$

Now substituting the righthand side of Eq. (5) for λ_{t+1} back into Eq. (4) and substituting the resulting expression for B_t in terms of a linear combination of L_t and λ_t,

$$\begin{bmatrix} L_{t+1} \\ \hline \lambda_{t+1} \end{bmatrix} = \left[\begin{array}{c|c} P & Q \\ \hline Y & Z \end{array}\right] \begin{bmatrix} L_t \\ \hline \lambda_t \end{bmatrix} + D \qquad (10)$$

where

$$P = [A + \beta(R - S'F_2S - F_1^*)^{-1} (S'F_2 + \beta_1F_1 - \beta'B_1^{-1}A_1)] . \qquad (11)$$

$$Q = \beta[(R - S'F_2S - F_1^*)^{-1} \beta'B_1^{-1}] . \qquad (12)$$

$$Y = -B_1^{-1}A_1 . \qquad (13)$$

$$Z = B_1^{-1} . \qquad (14)$$

$$D = \begin{bmatrix} -\beta(R - S'F_2S - F_1^*)^{-1} (S'F_2 + \beta_1F_1 - \beta'B_1^{-1}A_1)\bar{L} + C \\ \hline B_1^{-1}A_1L_1 \end{bmatrix} . \qquad (15)$$

These last results show that there is a discrete linear system whose terminal value L_T, after t steps, depends upon the value of the initial co-state vector λ_0.

If the α_i' as well as certain other model parameters are nonzero, the problem is well-posed in the sense that all the matrices that must be inverted are nonsingular.

If the transition matrix of Eq. (10) is

$$\phi \equiv \left[\begin{array}{c|c} P & Q \\ \hline Y & Z \end{array}\right] \tag{16}$$

then it is easy to show by induction on n that

$$\begin{bmatrix} L_{t+1} \\ ---- \\ \lambda_{t+1} \end{bmatrix} = \phi^{t+1} \begin{bmatrix} L_0 \\ ---- \\ \lambda_0 \end{bmatrix} + (I + \phi + \phi^2 + \ldots \phi^t)D \ . \tag{17}$$

Assuming that ϕ does not have a unit eigenvalue, Eq. (17) can be re-written as

$$\begin{bmatrix} L_{t+1} \\ ---- \\ \lambda_{t+1} \end{bmatrix} = \phi^{t+1} \begin{bmatrix} L_0 \\ ---- \\ \lambda_0 \end{bmatrix} + (I - \phi)^{-1}(I - \phi^{t+1})D \ . \tag{18}$$

In addition if ϕ is assumed to have no multiple eigenvalues, then

$$\phi = V \text{ diag } \mu_i V^{-1} \ , \tag{19}$$

where μ_i are the eigenvalues of ϕ and V is the complex matrix whose columns are the eigenvectors of ϕ. Therefore, Eq. (18) can be written

$$\begin{bmatrix} L_{n+1} \\ ---- \\ \lambda_{t+1} \end{bmatrix} = V \text{ diag } \mu_i^{n+1} \; V^{-1} \begin{bmatrix} L_0 \\ ---- \\ \lambda_0 \end{bmatrix} \qquad (20)$$

$$+ V \left[\text{diag } \frac{(1 - \mu_i^{n+1})}{(1 - \mu_i)} \right] V^{-1} D \; .$$

The decomposition of ϕ in Eq. (19) permits the potentially numerically unstable process of computing the powers of ϕ to be replaced by the more stable computation of powers of the eigenvalues of ϕ. Moreover, the spectral decomposition gives some insight into system stability in both the numerical and Lyapounov senses. Thus the system will be Lyapounov-stable and numerically stable if and only if the eigenvalues are all less than one in magnitude. This has not been the case with any of the models tested thus far. Hence, the methodology developed here will fail for a sufficiently large problem dimension or extent.

If the terminal state and co-state after T years are denoted as L_T and λ_T, then

$$\begin{bmatrix} L_T \\ ---- \\ \lambda_T \end{bmatrix} = \bar{\phi} \begin{bmatrix} L_0 \\ ---- \\ \lambda_0 \end{bmatrix} + \bar{D} \; , \qquad (21)$$

where

$$\bar{\phi} = V \text{ diag } \mu_i^T \; V^{-1} D \; ,$$

$$(22)$$

$$\bar{D} = V \text{ diag} \left[\frac{1 - \mu_i^T}{1 - \mu_i} \right] V^{-1} D \; .$$

Partitioning $\bar{\phi}$ into four blocks and $\bar{D}$ into upper and lower parts,

$$\begin{bmatrix} L_T \\ \hline \lambda_T \end{bmatrix} = \left[\begin{array}{c|c} E_{11} & E_{12} \\ \hline E_{13} & E_{22} \end{array}\right] \begin{bmatrix} L_0 \\ \hline \lambda_0 \end{bmatrix} + \begin{bmatrix} \bar{D}_u \\ \hline \bar{D}_1 \end{bmatrix} \tag{23}$$

or

$$L_T = E_{11}L_0 + E_{12}\lambda_0 + \bar{D}_u \tag{24}$$

or

$$\lambda_0 = E_{12}^{-1}(L_T - E_{11}L_0 - \bar{D}_u) \ , \tag{25}$$

provided E_{12} is nonsingular. (The nonsingularity of E_{12} is a sufficient but not necessary condition for the solution of Eq. (24) for λ_0.)

Given L_0, and λ_0 as determined from Eq. (25), it is possible to construct each state and co-state from the preceding according to

$$\begin{aligned} L_0 &= L_0 \ , \\ \lambda_0 &= \lambda_0 \ , \\ U_0 &= W_1[W_2(L_0 - L_1) + W_3\lambda_0] \ , \end{aligned} \tag{26}$$

$$L_{t+1} = AL_t + \beta B_t + C \ ,$$

$$\lambda_{t+1} = B_1^{-1}[A_1(L_t - L_T) + \lambda_t] \ , \tag{27}$$

$$B_{t+1} = W_1[W_2(L_{t+1} - L_T) + W_3\lambda_{t+1}] \ , \tag{28}$$

$$(t = 1, 2, \ldots, T-1) \ ,$$

where

$$W_1 = (R - S'F_2S - F_1^*)^{-1} \quad , \tag{29}$$

$$W_2 = S'F_2 + B_1F_1 - \beta'B_1^{-1}A_1 \quad , \tag{30}$$

$$W_3 = \beta'B_1^{-1} \quad . \tag{31}$$

Having sequentially constructed the state, co-state, and optimal bonuses as Eqs. (26) to (28) above, it is possible to construct, for each year t, an augmented and modified state vector, $\bar{L}_t$, such that

$$\bar{L}_t = \begin{bmatrix} K_1\beta B_{1,t} \\ \text{---------} \\ L_t + SB_n \end{bmatrix}_{m\times 1} \quad . \tag{32}$$

The rows of the modified G-table are given by

$$G_{t,j} = \begin{cases} \bar{L}_{t,j} \ , & j \le m \\ \lambda_{t,j} \ , & j = m+1, \ \ldots, \ 2m-1 \end{cases} \quad . \tag{33}$$

The bonus cost, $C_{B,t}$ of the controlled system in year t is given by

$$C_{B,t} = 1/2 \sum_i \beta_{i+4[\frac{m-1}{4}]} B^2_{i+4[\frac{m-i}{4}],t} \quad . \tag{34}$$

The penalty cost, $C_{P,t}$ of the controlled system is given by

$$C_{P,t} = -1/2(\bar{L}_t - \bar{L}_T)' \ F'(\bar{L}_t - \bar{L}_T) \quad , \tag{35}$$

where

$$\bar{L}_T = \begin{bmatrix} K_1 \\ ---- \\ \bar{L} \end{bmatrix} . \tag{36}$$

Total system cost, $C_{TOT,t}$ is given by

$$C_{TOT,t} = C_{B,t} + C_{P,t} . \tag{37}$$

It should be noted that

$$f(L_t, \lambda_t, U_t) = C_{TOT,t} . \tag{38}$$

The system Hamiltonian, H_t, in year t is given by

$$H_t = C_{TOT,t} + \lambda'_{t+1} L_{t+1} . \tag{39}$$

In contrast to the optimally controlled system, the uncontrolled system evolves according to

$$L^*_{t+1} = A^* L^*_t + [K_1, 0, \ldots, 0]' , \tag{40}$$

where

$$A^* = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \alpha_1 & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & \alpha_{m-1} & 0 \end{bmatrix} . \tag{41}$$

The only yearly cost associated with an uncontrolled system is the penalty cost, C_p^*, which is given by

$$C_p^* = -1/2(L_t^* - \bar{L}_T)' \; F(L_t^* - \bar{L}_T) \; . \tag{42}$$

The yearly gain accruing to the use of the optimally controlled over the uncontrolled system, g_t, is given by

$$g_t = C_{TOT,t} - C_{P,t}^* \; . \tag{43}$$

The total, g_{TOT}, is given by

$$g_{TOT} = \sum_{t=0}^{T} g_t \; .$$

III. PROGRAM OPERATING INSTRUCTIONS

FUNCTION

The DBMMP (Dynamic Bonus Management Model Program) computes the solution to a discrete, linear optimal control problem by using the discrete maximum principle. The problem is formulated in terms of matrix and vector algebra and a spectral decomposition of the basic state/co-state transition matrix plays a key role in the solution methodology. The cost function quadratic form matrix entries are pre-computed as functions of a set of economic model parameters.

ARCHITECTURE

The DBMMP consists of a main program and some 22 subroutines. All are coded in FORTRAN IV double precision for the IBM 370-158. Both real and complex arithmetic are used. Total source program length is about 1700 instructions.

STORAGE REQUIREMENTS AND TIMING

Core storage requirements for the program and its subroutines are dependent on problem dimensions. A sample problem characterized by eight year groups over a ten year manning horizon required 226K bytes of core at the GO step with object code produced by the FORTRAN G compiler. This problem was solved in a running time of 5.4 370-158 CPU seconds.

LIMITATIONS

Because of the problem's characteristic stability structure, its dimensions cannot be pushed too far. Numerical difficulties were encountered in the sample problem as the manning horizon was varied significantly beyond ten years.

INPUT

Table 2 lists DBMMP input parameter program names and definitions. The first two input parameters define the dimensions of the optimal

Table 2

DBMMP INPUT

Parameter Name	Definition	Units	Origin
N1	Number of year groups less one	Dimensionless (integer)	Card input
M	Problem extent	Years (integer)	Card input
T	Problem extent	Years (real)	Card input
ELZERO	Initial year group state vector at time zero	Men (real)	Card input
ELT	Terminal year group state vector augmented by one in dimension to include the first year group	Men (real)	Card input
D1	First distribution parameter	Dimensionless (real)	Card input
D2	Second distribution parameter	Dimensionless (real)	Card input
W2	Second term wages	Dollars (real)	Card input
RHO	Substitution parameter	Dimensionless (real)	Card input
ETA	Demand elasticity	Dimensionless (real)	Card input
EN	Returns to scale parameter	Dimensionless (real)	Card input
HP	Supply parameter vector	Dimensionless (real)	Card input
AA	No-bonus accession rate	Dimensionless (real)	Card input
ALPHA	No-bonus continuation rate	Dimensionless (real)	Card input

control problem. N1 is the number of different year groups. Currently N1 must always be set to 7. M, the problem extent or manning horizon, should not go much beyond 12 if numerical difficulties are to be avoided. T is just a real variable convenience. ELZERO is the N1 dimensional vector whose components are the number of men in year

groups 2, 3, ..., N1+1. ELT is the N1+1 dimensional vector of desired manpower level in each of the year groups 1, 2, ..., N1+1. The augmented first component of L_T is the constant input manpower level in year group 1. D1, the distribution parameter (δ_1); D2 (δ_2); and W2 are defined in Table 1. RHO is ρ, ETA is ε, EN is μ, HP is the vector of β_j', AA is the no-bonus accession rate equivalent to A_1, and ALPHA corresponds to α, all as defined in Table 1.

Table 3 defines the input data, card formats, and deck structure. Table 4 defines the Job Control Language (JCL) card requirements for executing a program load module whose Data Set Name is LATEST belonging to the partitioned data set PANBONUS.

OUTPUT

Because the current version of the DBMMP is a prototype research tool, much of its output is generated for purely diagnostic purposes. This section will describe only the most important output parameter values.

Table 5 shows a sample output echo of each of the basic input parameters. N1 is the reduced basic problem dimension, N is the row and column dimension that will be dealt with in the solution of the control problem, M is the problem extent in years, and T is the floating point problem extent.

The real N1 × 1 matrix LZRO is the initial vector of manpower level in year groups 2-8. The real N × 1 matrix LT is the desired final vector of manpower levels in year groups 1-8. The bottom row of the table is an echo of the economic model parameters previously described.

Table 6 is the N × 11 negation semi-definite matrix associated with the penalty cost for excesses or deficiencies in targeted manpower levels. It is computed as a function of LZERO, LT, and the parameters D1, D2, etc.

Table 7 is the N × N state/co-state transition matrix that moves the year groups and associated co-state of shadow prices from one year to the next. It is identical to the matrix ϕ defined in Eq. (16).

Table 3

INPUT CARD SPECIFICATIONS

Card \ Field	1	2	3	4	5	6	7	8	9	10
	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.	Format Cols.
1	N1 I6, 1-6	M I6, 7-12	T F20.16, 13-32							
2	ELZERO(1) F8.0, 1-8	ELZERO(2) F8.0, 9-16	ELZERO(3) F8.0, 17-24	ELZERO(4) F8.0, 25-32	ELZERO(5) F8.0, 33-40	ELZERO(6) F8.0, 41-48	ELZERO(7) F8.0, 49-56	ELZERO(8) F8.0, 57-64	ELZERO(9) F8.0, 65-72	ELZERO(10) F8.0, 73-80
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\frac{N1}{10}+2$ if 10|N1[a]	ELT(1)	ELT(2)	ELT(3)	ELT(4)	ELT(5)	ELT(6)	ELT(7)	ELT(8)	ELT(9)	ELT(10)
$\frac{N1}{10}+3$ if 10|N1	F8.0, 1-8	F8.0, 9-16	F8.0, 17-24	F8.0, 25-32	F8.0, 33-40	F8.0, 41-48	F8.0, 49-56	F8.0, 57-64	F8.0, 65-72	F8.0, 73-80
$\frac{N}{5}+3$ if 10|N1	D1	D2	W2	RHO	ETA	EN	HP(1)	HP(5)	HP(9)	HP(14)
$\frac{N1}{5}+4$ if 10|N1	F8.0, 1-8	F8.0, 9-16	F8.0, 17-24	F8.0, 25-32	F8.0, 33-40	F8.0, 41-48	F8.0, 49-56	F8.0, 57-64	F8.0, 65-72	F8.0, 73-80
$\frac{N1}{5}+4$ if 10|N1	ALPHA(1)	ALPHA(2)	ALPHA(3)	ALPHA(4)	ALPHA(5)	ALPHA(6)	ALPHA(7)	ALPHA(8)	ALPHA(9)	ALPHA(10)
$\frac{N1}{5}+5$ if 10|N1	F8.0, 1-8	F8.0, 9-16	F8.0, 17-24	F8.0, 25-32	F8.0, 33-40	F8.0, 41-48	F8.0, 49-56	F8.0, 57-64	F8.0, 65-72	F8.0, 73-80
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

[a] 10|N1 means 10 does not divide N1.

Table 4

PANBONUS JCL

```
//S4610#01 JOB (3191,60,53),"FEBRUARY",CLASS=F
//GO EXEC PGM=LATEST,REGION=230K
//STEPLIB DD DSN=S.S4610.A3191.PANBONUS,DISP=SHR
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT06F001 DD SYSOUT=A
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//GO.FT10F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//GO.FT11F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//GO.FT12F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//GO.FT13F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//GO.SYSIN DD *
    7     8          8.0
  350.   250.   250.   160.   170.   180.   220.
  323.   307.   292.   278.   200.   200.   200.   200.
  .45   .55    10000.  .25    .68    .032   .005   323.
  .95   .95    .95     .72    1.0    1.0    1.0    .95
/*
/*
```

Table 8, (M+1) × (2N-1), is a tableau of the year-by-year optimal state/co-state solutions. The lefthand row numbers should be reduced by one to give the appropriate year. The first eight entries in each row are the manpower levels in each year group i, i = 1, 2, ..., 8, and the last seven entries are the corresponding co-state variables.

Table 9 summarizes the costs associated with optimal system control and compares them with the costs incurred without control. To be precise, however, the year in the second column at the left should be reduced by one. PENALTY COST is the quadratic component of the cost associated with year group deviations from desired final manpower levels. BONUS COST is the quadratic cost component associated with bonus payments. HAMILTONIAN is the discrete system Hamiltonian. BONUS 1 and BONUS 5 are the optimal bonuses paid to men entering the first and fifth year groups. PENALTY COST* is the pure quadratic penalty cost corresponding to a no-bonus policy. NET GAIN is the difference in the cost of the uncontrolled and optimally controlled systems. The last line of the table totals the various cost components.

Table 5

DBMMP INPUT PARAMETER ECHO

```
NI = 7N = 14 M =    8  T =  8.00000000


   REAL MATRIX  LZRO
 0.35000D 03
 0.25000D 03
 0.25000D 03
 0.16000D 03
 0.17000D 03
 0.18000D 03
 0.22000D 03


   REAL MATRIX  LT
 0.32300D 03
 0.30700D 03
 0.29200D 03
 0.27800D 03
 0.20000D 03
 0.20000D 03
 0.20000D 03
 0.20000D 03

D1     D2      W2     RHO     ETA    HP1     HP5       AA      --------------------ALPHAS-------------------- EN
0.45  0.55 10000.0  0.250   0.680  0.032   0.005323.000   0.950  0.950  0.950  0.720  1.000  1.000  1.000  0.950
```

Table 6

F-MATRIX OF PENALTY COSTS

F TABLE

-5.47814	-5.47814	-5.47814	-5.47814	-0.69801	-0.69801	-0.69801	-0.69801
-5.47814	-5.47814	-5.47814	-5.47814	-0.69801	-0.69801	-0.69801	-0.69801
-5.47814	-5.47814	-5.47814	-5.47814	-0.69801	-0.69801	-0.69801	-0.69801
-5.47814	-5.47814	-5.47814	-5.47814	-0.69801	-0.69801	-0.69801	-0.69801
-0.69801	-0.69801	-0.69801	-0.69801	-17.04121	-17.04121	-17.04121	-17.04121
-0.69801	-0.69801	-0.69801	-0.69801	-17.04121	-17.04121	-17.04121	-17.04121
-0.69801	-0.69801	-0.69801	-0.69801	-17.04121	-17.04121	-17.04121	-17.04121
-0.69801	-0.69801	-0.69801	-0.69801	-17.04121	-17.04121	-17.04121	-17.04121

Table 7

STATE/CO-STATE TRANSITION MATRIX

REAL MATRIX R

-0.9500D 00	-0.9500D 00	-0.9500D 00	-0.2319D 02	-0.2319D 02	-0.2319D 02	-0.2319D 02	0.0	0.0	0.0
-0.1160D 00	0.0	0.0	-0.1361D 01						
0.9500D 00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0						
0.0	0.9500D 00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0						
0.0	0.0	0.7200D 00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0						
-0.5963D-18	-0.5963D-18	-0.5963D-18	0.1000D 01	0.1388D-16	0.1388D-16	0.1388D-16	0.0	0.0	0.0
0.5000D-02	0.0	0.0	0.8674D-18						
0.0	0.0	0.0	0.0	0.1000D 01	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0						
0.0	0.0	0.0	0.0	0.0	0.1000D 01	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0						
-0.3289D 02	-0.3289D 02	-0.3289D 02	-0.9431D 03	-0.9431D 03	-0.9431D 03	-0.9431D 03	0.0	0.0	0.0
-0.4716D 01	0.0	0.0	-0.5539D 02						
0.2220D-15	0.2220D-15	0.2220D-15	-0.1400D 03	-0.1400D 03	-0.1400D 03	-0.1400D 03	0.1053D 01	0.0	0.0
-0.7002D 00	0.0	0.0	-0.8261D 01						
0.2220D-15	0.2220D-15	0.2220D-15	-0.1400D 03	-0.1400D 03	-0.1400D 03	-0.1400D 03	0.0	0.1053D 01	0.0
-0.7002D 00	0.0	0.0	-0.8261D 01						
0.0	0.0	0.0	-0.1848D 03	-0.1848D 03	-0.1848D 03	-0.1848D 03	0.0	0.0	0.1389D 01
-0.9239D 00	0.0	0.0	-0.1090D 02						
-0.4163D-16	-0.4163D-16	-0.4163D-16	-0.1332D-14	-0.1332D-14	-0.1332D-14	-0.1332D-14	0.0	0.0	0.0
0.1000D 01	0.0	0.0	-0.1000D 01						
0.4163D-16	0.4163D-16	0.4163D-16	0.8882D-15	0.8882D-15	0.8882D-15	0.8882D-15	0.0	0.0	0.0
0.1388D-16	0.1000D 01	0.0	-0.1000D 01						
0.4163D-16	0.4163D-16	0.4163D-16	0.8882D-15	0.8882D-15	0.8882D-15	0.8882D-15	0.0	0.0	0.0
0.1388D-16	0.0	0.1000D 01	-0.1000D 01						

Table 8

TABLEAU OF OPTIMAL SOLUTIONS

G TABLE															
1	321.0	350.0	250.0	250.0	179.1	170.0	180.0	220.0	84.8	1364.5	2331.2	3813.6	3473.1	2186.1	888
2	323.3	305.0	332.5	237.5	194.8	179.1	170.0	180.0	-269.2	-115.2	1231.8	2968.0	2925.4	2584.9	1297
3	322.6	307.1	289.7	315.9	179.1	194.8	179.1	170.0	-55.8	-349.2	-187.1	1624.0	1670.1	1627.5	1287
4	323.7	306.5	291.8	275.2	227.1	179.1	194.8	179.1	134.9	88.7	-220.2	-65.4	337.0	383.0	340
5	324.0	307.6	291.2	277.2	196.4	227.1	179.1	194.8	-6.2	111.6	62.9	-346.0	-405.9	-3.5	42
6	324.2	307.8	292.2	276.6	200.0	196.4	227.1	179.1	30.0	-8.9	115.0	84.2	-388.5	-448.4	-46
7	323.5	308.0	292.4	277.6	200.0	200.0	196.4	227.1	47.6	38.2	-2.8	168.5	130.2	-342.5	-402
8	323.2	307.4	292.6	277.8	200.0	200.0	200.0	196.4	44.1	76.3	66.4	30.7	570.9	532.6	59
9	323.0	307.0	292.0	278.0	200.0	200.0	200.0	200.0	8.0	49.2	83.1	96.0	-29.2	511.0	472

Table 9

OPTIMAL SYSTEM VS UNCONTROLLED SYSTEM COSTS

EXTENT	YEAR	PENALTY COST	BONUS COST	TOTAL COST	HAMILTONIAN	BONUS 1	BONUS 5	PENALTY COST*	NET GAIN
8.00	1.00	0.254315D 05	0.364193D 05	0.618508D 05	0.188546D 07	-0.615236D 02	0.381359D 04	0.417510D 05	-0.200998
8.00	2.00	0.494338D 05	0.220233D 05	0.714571D 05	0.877652D 06	0.952150D 01	0.296796D 04	0.102922D 06	0.314654
8.00	3.00	C.520081D 05	0.659571D 04	0.586038D 05	0.129129D 06	-0.118375D 02	0.162400D 04	0.125937D 06	0.673334
8.00	4.00	0.342254D 04	0.191857D 02	0.344172D 04	-0.107907D 06	0.230443D 02	-0.653884D 02	0.326380D 05	0.291963
8.00	5.00	0.530939D 02	0.314467D 03	0.367561D 03	-0.131498D C6	0.308227D 02	-0.345986D 03	0.273420D 03	-0.941409
8.00	6.00	0.640188D 02	0.419451D 02	0.105964D 03	-0.739272D 05	0.389224D 02	0.841566D 02	0.163289D 04	0.152692
8.00	7.00	0.475695D 04	0.756685D 02	0.483262D 04	0.288140D 06	0.170533D 02	0.168541D 03	0.151023D 05	0.103497
8.00	8.00	0.107708D 03	0.275054D 01	0.110459D 03	0.249915D 06	0.493421D 01	0.307310D 02	0.171684D 04	0.160638
8.00	9.00	0.402192D-14	0.0	0.402192D-14	0.0	0.0	0.0	0.614610D 02	0.614610
TOTAL		0.135278D 06	0.654924D 05	0.200770D 06				0.322116D 06	0.121346

IV. PROGRAM DESCRIPTION

PROGRAM FUNCTION

The Dynamic Bonus Management Model Program is designed to determine an optimal policy for the payment of yearly bonuses to military personnel in a number of different "year groups." The bonuses are treated as the control variables in a discrete linear optimal control system in which the components of the current state vector are the number of men in each of a specified number year group. The objective of the optimal control policy is to drive the system from a given initial state in year zero to a prescribed terminal state over a specified number of years, while minimizing a given cost function. This cost function is the summation over each year of a positive semi-definite quadratic form in both the control variables and the difference between the year's state and the desired terminal state. The quadratic form itself is computed as a function of a set of system-characterizing economic model parameters and the terminal state.

The solution for the optimal policy is derived as a consequence of the discrete version of the maximum principle. Essentially, it reduces to the construction of a discrete linear system of twice the dimension of the year-group state and the subsequent determination of an unknown initial co-state vector that will yield the prescribed terminal state.

In addition to computing the sequence of yearly bonuses and associated costs for the optimal bonus policy, the program also performs a spectral analysis of the system state/co-state transition matrix. It also produces a net-gain comparison between the optimal policy and the no-control policy of zero bonuses.

PROGRAM ARCHITECTURE

The DBMMP is organized into a FORTRAN IV main program and some 22 subroutines. Communication between main and subroutine and from subroutine-to-subroutine is accomplished by the use of an extensive common block structure and, where appropriate, calling sequence parameters.

The main program is responsible for calling the data input and problem setup subroutine, PROGO, solving for the required initial co-state vector, sequentially computing the optimal bonuses, and calling the computational and analysis subroutine COSTS and OUTPUT to print out the states, co-states, bonuses, associated costs, system Hamiltonians, uncontrolled system costs, and net gains.

The subroutines are responsible for I/O and the linear algebra and eigenanalysis functions that solve the optimal control problem.

PROGRAM INPUT

Program input is handled entirely by the subroutine PROGO. For detailed input data card specifications see Sec. III.

PROGRAM OUTPUT

Program outputting functions are shared among the main program and a number of major subroutines. In its current research tool state, much of the program's output is diagnostic and the reader need not be concerned with it. The important outputs are: the echo of the input parameters; the negative, semi-definite cost quadratic form, F; the system state/co-state transition matrix, R; the array of modified year group levels (G TABLE); and the final array of penalty cost, bonus cost, total cost, system Hamiltonian, yearly bonuses, penalty cost of the uncontrolled system, net gain, and cumulative cost and net gain.

PROGO echoes out input parameters and prints out the matrix of the cost quadratic form. The PROGO-called subroutine, COMPUT, writes out the system state/co-state transition matrix. The main program prints out the transition matrix eigenvalues. The subroutine COSTS writes out the two parts of the A-Matrix, and OUTPUT is responsible for the unmodified L and ELAMB table output. COSTS also handles the outputting of cost/benefit performance parameters and the yearly Hamiltonian.

For detailed definitions of the output parameters, see the example of output given in Sec. III and the description of the relevant outputting subroutine.

PROGRAM COMPUTATIONAL SEQUENCE

Program operation begins with setting the maximum of the size of the problem dimension to 40. PROGO is then called to read input parameters and set up the various matrices and vectors that characterize the problem. PROGO calls COMPUT to generate the uncontrolled system state transition matrix and cost quadratic form matrix. From these, together with the control variable matrix multiplier, PROGO generates the state/co-state transition matrix. It also sets the complex part of the transition matrix to zero. When PROGO returns control to the main program, various loop limits are set and the real part of the state/co-state transition matrix is saved. Next, CBAL is called to balance the state/co-state transition matrix, COMHES is called to compute the upper-Hessenburg matrix similar to it, COMLR2 finds its complex eigenvalues, and CBABK2 back-transforms the corresponding complex eigenvectors to those of the original state/co-state matrix. The complex matrix of these eigenvectors is used subsequently to simplify and stabilize the required computation of the partial sum (up to "T" years) of a geometric power series in the state/co-state transition matrix.

The resultant geometric series partial sum matrix is then partitioned along with initial and desired final states to determine the required initial value of the co-state vector. Once this initial value is determined, the main program computes each year group state vector, co-state vector, and optimal bonuses recursively from the previous year's values. This process also permits the sequential construction of each of the output modified state, co-state, and cost analysis tabular entries. The modified state table is used both as a user output and computational convenience.

PROGRAM FLOW DIAGRAM AND SUBROUTINE CALLING MAP

To supplement this prose description of the program computational sequence, Fig. 1 presents a flow diagram of the DBMMP main program. For a definition of the mathematical symbols and equations development, refer to Sec. II.

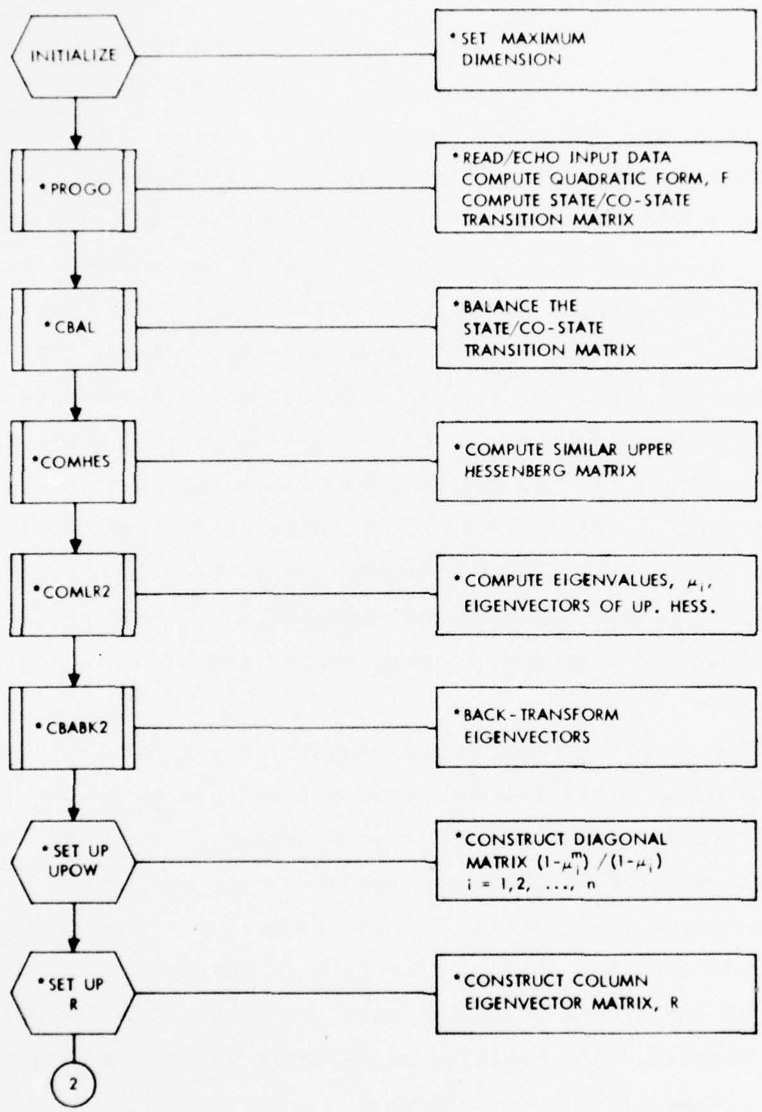

Fig. 1--Flow diagram of the DBMMP main program

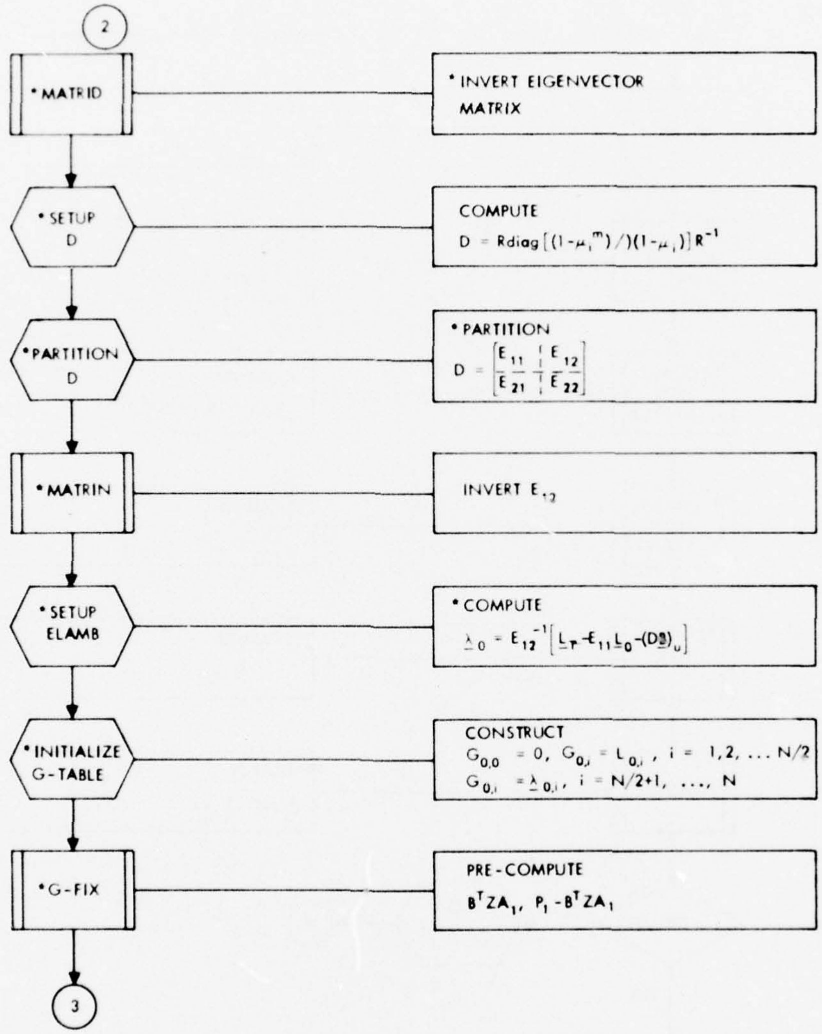

Fig. 1--continued

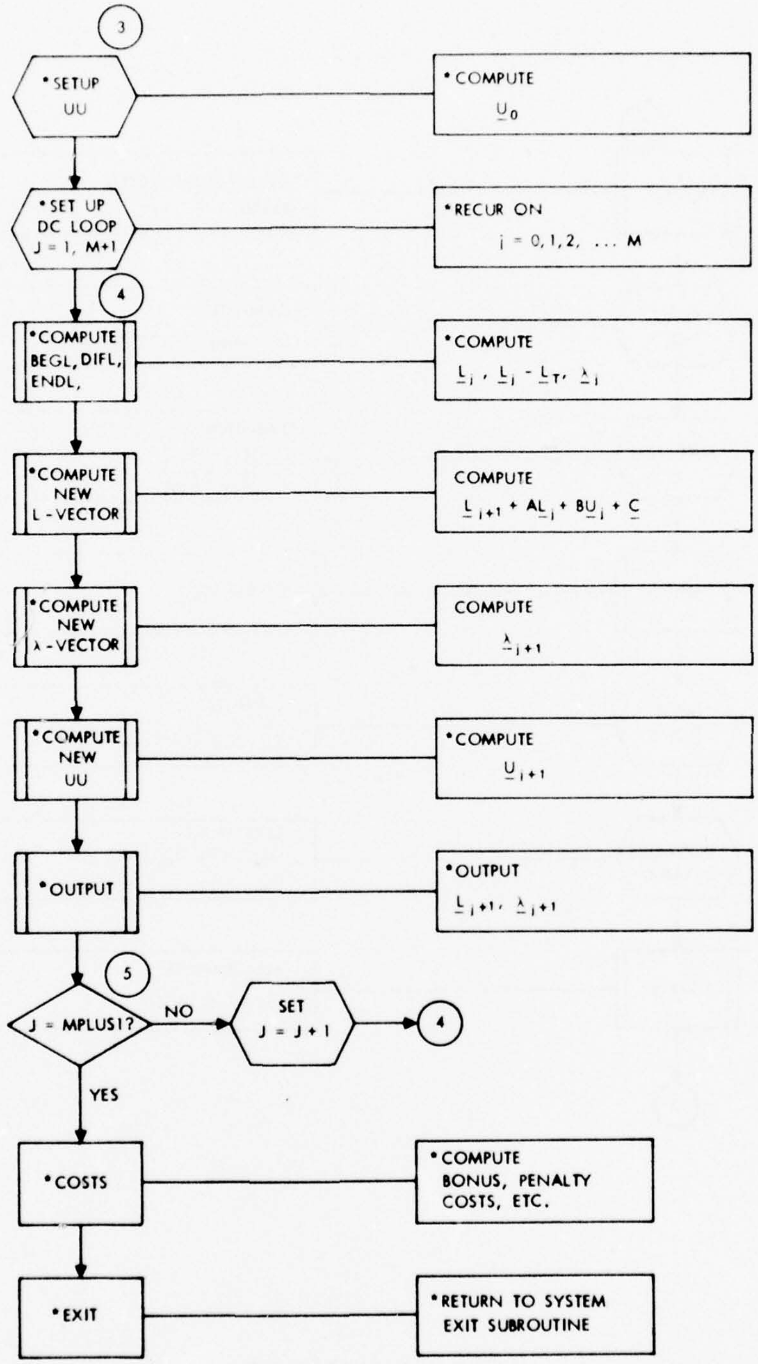

Fig. 1--continued

As a further aid to understanding the program's functional flow, Fig. 2 gives a subroutine calling map, which is a compact representation of the program's functional flow. Every subroutine appears in order of its first use by the calling main program or subroutine. But the subroutines do not appear again if they are called repeatedly by the calling subroutine or main program.

SUBROUTINE DESCRIPTION

In the sequel, each subroutine's full input is described in terms of explicit calling sequence structure and those parameters that are implicitly accessed through COMMON.

Subroutine PROGO (T,M,JK)

PROGO's primary functions are total program input and construction of problem-characterizing matrices and vectors. It reads in the dimension of the problem (number of different year groups beyond the first year group), the problems extent (number of transitional years), the initial state vector, the desired terminal state vector, and the problem characteristic economic parameters. From these are computed the entries of the cost function quadratic form. In addition, PROGO initializes to zero the real and imaginary parts of the state/co-state transition matrix whose dimension is twice the dimension of the number of year groups. The entries of the matrix are filled in by the PROGO-called subroutine COMPUT.

Refer to Table 3 for PROGO's input parameter names and definitions. All PROGO input originates by card input.

Table 10 lists the set of PROGO output parameters, definitions, and units.

On initiation, PROGO sets the maximum allowable state/co-state dimension to 40. Next it reads N1, the basic problem dimension; M, the integer problem extent; and T, the real floating point problem extent. It then sets to zero intermediate storage and the real and imaginary parts of the sta /co-state transition matrix. Then, the initial and desired final state vectors are read in. The economic model parameters are read in along with the year group transition

```
MAIN
  PROGO
    PRINTR
    COMPUT
      PRINTR
      MATMAT
      MATRIN
      TRANSP
      MATVEC
    CBAL
    COMHES
    COMLR2
    CBABK2
    PRINTC
    MATRID
    MATDIA
    MATMTC
    MATVEC
    MATRIN
    GFIX
      MATMAT
    MATMAT
    OUTPUT
    COSTS
      GFIX
      STAR
    EXIT (System Subroutine)
```

Fig. 2--DBMMP calling map

Table 10

PROGO OUTPUT

Parameter	Definition	Units
F	Negative semi-definite cost quadratic form	Dollars/men^2 (real)
A	State transition matrix	Dimensionless (real)
BB	Control variable matrix multiplier	Men/dollars (real)
B	Constant vector in state/co-state transition equation	[Men, men/dollars] (real)
AR	Real part of the state/co-state transition matrix	(a)
AI	Imaginary part of the state/co-state transition matrix	(a)
BTZ	Auxiliary matrix for computation of stagewise bonuses	Men/dollars (real)

[a] Both AR and AI have the units structure

$$\left[\begin{array}{c|c} \text{Dimensionless} & \text{Dollars} \\ \hline \text{Dollars}^{-1} & \text{Dimensionless} \end{array}\right].$$

parameters. Subsequently, the cost quadratic form matrix entries are computed as a function of the economic model parameters D1, D2, W2, ..., etc. Once F has been computed, PROGO branches to COMPUT for the computation of the last six of PROGO's output parameters, A, BB, B, AR, AI, BTZ.

Subroutine PRINTR(OUTMAT,N,M,MA,K)

PRINTR prints out a real matrix according to a number of variable formats. The particular format is selected on the basis of an input parameter. Table 11 lists PRINTR input parameters, definitions, and origins.

Table 11

PRINTR INPUT

Parameter	Definition	Units	Origin
OUTPUT	Location of the first entry of the matrix to be printed	Dimensionless (integer)	Calling program
N	Row dimension of the matrix	Dimensionless (integer)	Calling program
M	Column dimension of the matrix	Dimensionless (integer)	Calling program
MA	Maximum row dimension	Dimensionless (integer)	Calling program
K	Heading specification	Dimensionless (integer)	Calling program

PRINTR's output is row-by-row entries of the input matrix, OUTMAT, ten row entries of D12.4 format per line.

Subroutine COMPUT

COMPUT sets up the basic matrices involved in the optimization problem. Table 12 lists COMPUT's input parameters, definitions, units, and origins. Table 13 lists its output parameters, definitions, and units.

COMPUT begins its operation by setting the maximum dimension to N1, the basic overall problem dimension. It then sets to zero the effective constant input vector, CC, and the state transition matrix, A, and several auxiliary matrices S, BB, ST, and P1. Nonzero entries are next computed for the state transition matrix, A, the control variable matrix, BB, and the effective reduced dimension constant input, CC. The positive definite quadratic form in the control variables, RR, is then set to its initial value. Subsequently, the auxiliary matrix, F_1, is constructed; and the reduced order cost quadratic format, F_2, is set up as the lower N1 × N1 diagonal block of the original quadratic form. In the final sequence of computations, each of the four N1 × N1 blocks of the real part of the state/co-state transition

Table 12

COMPUT INPUT

Parameter	Definition	Units	Origin
N1	Problem dimension	Dimensionless (integer)	PROGO
ALPHA	State transition parameters	Dimensionless (integer)	PROGO
HP	Supply parameter vector	Dollars^{-1} (real)	PROGO
AA		Dimensionless (real)	PROGO
F	Cost quadratic form	Dollars/men^2 (real)	PROGO

Table 13

COMPUT OUTPUT

Parameter	Definition	Units
AR	The real part of the state/co-state transition matrix	(See Sec. II)
AI	The zero imaginary part of the state/co-state transition matrix	(See Sec. II)
B	The constant state/co-state input vector	(Men, men/dollar)
BB	The control variables multiplier matrix	Men/dollar
BTZ	Auxiliary matrix for sequential bonus computation	$\text{Men}^2\text{/dollar}$ (real)

matrix, AR, is computed along with the constant state/co-state transition-input vector, B, and the auxiliary bonus computation matrix BTZ. For a detailed understanding of the meaning of *PROGO*'s operation and symbols see Appendix A.

MATMAT, the DBMMP System Real Matrix Multiplication Subroutine

Table 14 shows MATMAT's input parameters, definitions, and origins. A "______" in the units column means the units are determined by the particular input.

Table 14

MATMAT INPUT

Parameter	Definition	Units	Origin
AMATIN	Pro-multiplier matrix	______ (real)	Calling program
BMATIN	Post-multiplier matrix	______ (real)	Calling program
NI	Row dimension of AMATIN	Dimensionless (integer)	Calling program
NK	Column dimension of AMATIN = row dimension of BMATIN	Dimensionless (integer)	Calling program
NJ	Column dimension of BMATIN	Dimensionless (integer)	Calling program

The output of MATMAT, OUTMAT, is the NI × NJ dimensional product AMATIN * BMATIN.

On entrance, the product matrix, OUTMAT, is set to zero. Then the product, according to the usual matrix multiplication rules, is

$$c_{ik} = \sum_{j=1}^{m} a_{ij} b_{jk} ,$$

m being the common column dimension of (a_{ij}) and the row dimension of (b_{jk}).

Subroutine MATRIN, the DBMMP System Real Matrix Inversion Subroutine

MATRIN's input parameters, definitions, units, and origins are given in Table 15.

Table 15

MATRIN INPUT

Parameter	Definition	Units	Origin
A	Matrix to be inverted	______ (real)	Calling program
N1	Maximum row dimension of A	______ (real)	Calling program
N	Actual row dimension of A	______ (real)	Calling program
B	Matrix of righthand side vectors in the problem AX=B	______ (real)	Calling program
M1	Row dimension of B	Dimensionless (integer)	Calling program
M	Column dimension of B	Dimensionless (integer)	Calling program
IPIVOT	Internally used one-dimension integer array	Dimensionless (integer)	Calling program
INDEX	Internally used two-dimensional integer array	Dimensionless (integer)	Calling program
ISING	Singular/nonsingular flag	Dimensionless (integer)	Calling program
DETERM	Determinant of A	______ (real)	Calling program

MATRIN's output, A^{-1}, is stored in A. A is destroyed and must be saved before the call to MATRIN if it is to be used again. MATRIN uses Gauss-Jordan pivotal elimination with full pivoting. At each of NI stages, the element of maximal absolute value is selected from among those row and column entries that have not previously been used as a pivot element. Rows are then permuted to bring the pivotal element

into the diagonal position and the column entries above and below the new maximal element are annihilated by appropriate elementary row operations. The row interchanges are kept track of and at the end of the last stage the columns of the inverse are permuted inverse to the original row transpositions to unscramble them.

TRANSP, the DBMMP System Matrix Transposition Subroutine

TRANSP's input parameters, definitions, units, and origins are given in Table 16. Its output, XOUT, is the NJ × NI transpose of XIN.

Table 16

TRANSP INPUT

Parameter	Definition	Units	Origin
XIN	The matrix to be transposed	______(real)	Calling program
NI	Row dimension of XIN	Dimensionless (integer)	Calling program
NJ	Column dimension of XIN	Dimensionless (integer)	Calling program

MATVEC, the DBMMP System Matrix, Vector Multiplication Subroutine

MATVEC's input parameters, definitions, units, and origins are given in Table 17. MATVEC's output is the N dimensional real vector

Table 17

MATVEC INPUT

Parameter	Definition	Units	Origin
AMATIN	Pre-multiplying matrix	______(real)	Calling program
VECTIN	Post-multiplying vector	______(real)	Calling program
NI	Maximum row dimension of AMATIN	Dimensionless (integer)	Calling program
N	Row dimension of A	Dimensionless (integer)	Calling program

VECOUT. After setting the product vector, VECOUT, to zero, MATVEC computes the matrix-vector product according to the usual rule:

$$b_i = \sum_{j=1}^{n} a_{ij} x_j \ .$$

Subroutine CBAL(NM,N,AR,AI,LOW,IGH,SCALE)

CBAL is used to pre-condition the complex state/co-state transition matrix, AR + iAI, (AI=0), balancing it and isolating its eigenvalues whenever possible. Balancing is performed by elementary row and column interchanging and scaling. CBAL is part of the Argonne National Laboratory EISPACK eigenvalue/eigenvector package. Real arithmetic is used throughout CBAL.

CBAL's input parameter definition, units, and origins are given in Table 18. CBAL's output parameter definition and units are shown in Table 19.

Table 18

CBAL INPUT

Parameter	Definition	Units	Origin
NM	Maximum row dimension of matrix to be balanced	Dimensionless (integer)	Main
N	The order of the matrix	Dimensionless (integer)	Main
AR	Real part of the matrix	See Sec. II	Main
AI	Imaginary part of matrix	See Sec. II	Main

Subroutine COMHES(MA,N,AR,AI,LØ,IGH,INT)

DBMMP uses COMHES to reduce the balanced matrix output of CBAL to upper Hessenberg form by stabilized elementary similarity transformations. COMHES input parameter definitions, units, and origins are shown in Table 20.

Table 19

CBAL OUTPUT

Parameter	Definition	Units
AR	The real part of the balanced matrix	See Sec. II
AI	The imaginary part of the balanced matrix	See Sec. II
LØ	Indices specifying location of zero block in the balanced matrix	Dimensionless (integer)
IGH	Indices specifying location of zero block in the balanced matrix	Dimensionless (integer)
SCALE	N dimensional vector of scaling information	Dimensionless (real)

Table 20

COMHES INPUT

Parameter	Definition	Unit	Origin
MA	Matrix row dimension of the balanced matrix	Dimensionless (integer)	Main program
N	Row dimension of the balanced matrix	Dimensionless (integer)	Main program
LOW, IGH	Balanced matrix zero block bounding indices	Dimensionless (integer)	Main program by CBAL
AR	Real part of the balanced matrix	See Sec. II	Main program by CBAL
AI	Real part of the balanced matrix	See Sec. II	Main program by CBAL

Output

COMHES's output parameter definitions and units are given in Table 21.

Table 21

COMHES OUTPUT

Parameter	Definition	Units
AR	Real part of the upper Hessenberg matrix	See Sec. II
AI	Imaginary part of the upper Hessenberg matrix	See Sec. II
INT	Integer array containing row-columns interchange information	Dimensionless (integer)

Subroutine COMLR2(MA,N,LØ,IGH,INT,AR,AI,WR,WI,ZR,ZI,IERR)

COMLR2 computes the eigenvalues and eigenvectors of the reduced Hessenberg matrix output of COMHES. COMLR2's input parameter definitions, units, and origins are shown in Table 22. COMLR2's output parameter definitions and units are given in Table 23.

Subroutine CBABK2(MA,N,LØ,IGH,SCALE,N,ZR,ZI)

CBABK2 back-transforms the eigenvectors of the upper Hessenberg matrix obtained from the output of COMLR2 to the eigenvectors of the original unbalanced state/co-state transition matrix.

CBABK2's parameter definition, units, and origins are shown in Table 24. On output ZR and ZI are the real and imaginary parts of the eigenvectors of the original, complex, unbalanced state/co-state matrix.

PRINTC, the Complex Analog of PRINTR

PRINTC prints out a complex matrix preceded by an accompanying variable format. Its input parameter definitions, units, and origins are shown in Table 25. PRINTC's output is variable header 10D12.4 format with five complex entries (real, imaginary) per output line.

Table 22

COMLR2 INPUT

Parameter	Definition	Units	Origin
MA	Maximum dimension of the Hessenberg matrix	Dimensionless (integer)	Main program by COMHES
N	Dimension of the Hessenberg matrix	Dimensionless (integer)	Main program by COMHES
LØW	Zero block boundary indices determined by CBAL		
IGH	Zero block boundary indices determined by CBAL		
INT	Vector of row-column permutations		
AR	Real part of the upper Hessenberg matrix	See Sec. II	
AI	Imaginary part of the upper Hessenberg matrix	See Sec. II	

Table 23

COMLR2 OUTPUT

Parameter	Definition	Units
WR	Real part of the eigenvalues of the upper Hessenberg matrix	Dimensionless (real)
WI	Imaginary part of the eigenvalues of the upper Hessenberg matrix	Dimensionless (real)
ZR	Real part of the eigenvector of the upper Hessenberg matrix	[Men, men/dollar] (real)
ZI	Imaginary part of the upper Hessenberg matrix	[Men, men/dollar] (real)
IERR	Error indicator: 0 if all eigenvalues determined, j if jth eigenvalue not determined	Dimensionless (integer)

Table 24

CBABK2 INPUT

Parameter	Definition	Units	Origin
MA	Maximum dimension of original unbalanced matrix	Dimensionless (integer)	Main program
N	Dimension of the matrix		Main program
LOW IGH	Zero block boundary indices		Main program by CBAL
SCALE	Row-column formulation/ scaling information	______(real)	Main program by CBAL
ZR	Real part of the eigen- vector of the reduced Hessenberg matrix	[Men, men/dollar]	Main program by COMLRZ
ZI	Imaginary part of the eigenvector	[Men, men/dollar]	Main program by COMLRZ

Table 25

PRINTC INPUT

Parameter	Definition	Units	Origin
OUTMAT	Complex matrix to to be printed out	______(complex)	Main program
N	Row dimension of OUTMAT	______(real)	Main program
M	Column dimension of OUTMAT	______(real)	Main program
MA	Maximum row dimension of OUTMAT	______(real)	Main program

MATRID, DBMMP's Complex Matrix Inversion Subroutine Analog of MATRIN

See subroutine MATRIN for subroutine calling sequence, I/O parameter definitions, computational sequence, etc.

Subroutine MATDIA (AMATIN,DIAGIN,N,N1,OUTMAT)

MATDIA post-multiplies a complex square matrix by a complex diagonal matrix. MATDIA's input parameter definitions, units, and origins are shown in Table 26.

Table 26

MATDIA INPUT

Parameter	Definition	Units	Origin
AMATIN	Pre-multiplier matrix	______ (complex)	Main program
DIAGIN	Vector of post-*multiplier matrix* diagonal entries	______ (complex)	Main program
N	Dimension of AMATIN	Dimensionless (integer)	Main program
N1	Maximum row dimension	Dimensionless (integer)	Main program

MATDIA's output is the complex product AMATIN* diag (DIAGIN(J)). Each entry in the jth column of AMATIN is multiplied by DIAGIN(J).

MATMTC, the DBMMP Systems Complex Matrix Multiplication Subroutine

MATMTC functions analogously to MATMAT. For calling sequence parameter definitions, etc., see section MATMAT.

Subroutine GFIX (M,ITIME)

GFIX modifies the state/co-state versus year table so that it can be used by the costs subroutine. It computes and inserts the leading year group and entry and modifies the fifth year group entry. In addition, GFIX computes the matrix constant used in the sequential computation of the optimum bonuses. GFIX input parameter definitions,

units, and origins are shown in Table 27. GFIX output parameter definitions and units are shown in Table 28.

Table 27

GFIX INPUT

Parameter	Definition	Units	Origin
M	Problem extent	Years (integer)	Main program
ITIME	Year flag	Dimension (integer)	Main program
N	Total problem dimension	Dimension (integer)	Main program
BTZ	Auxiliary matrix	Men^2/dollar (real)	COMPUT
A1	Auxiliary matrix	Dimension (real)	COMPUT
P1	Auxiliary matrix	Dollars/men^2 (real)	COMPUT
HP	Supply parameter vector	$Dollars^{-1}$ (real)	PROGO
UU	*Bonus values*	Dollars (real)	Main program

On entrance GFIX checks the ITIME input flag to determine whether it is the final time through. If ITIME = 1 and it is the first time through, GFIX computes the looping indices NPLUS1 and MPLUS1 and the auxiliary matrices BTZA and TERM1, which will be used by the main program for sequential bonus computation and returns to the calling program MAIN. Otherwise, looping over each year, it modifies the first and fifth entries of the G-table for the given year specified by the value of M by adding linear multiples of the optimal bonuses for that

Table 28

GFIX OUTPUT

Parameter	Definition	Units
BTZA	Auxiliary matrix	Men^2/dollar (real)
TERM1	Auxiliary matrix	Dimensionless (real)
G	Modified G-table	[Men, men/dollar] (real)

year. At the conclusion of the loop, GFIX returns to the calling program, which, in this case, is the subroutine COSTS.

Subroutine OUTPUT(JK,JJ,J2,J3,J4,MMM)

OUTPUT prints out both the identification headers for L-table and ELAMB-table and the tabular entries themselves.

OUTPUT input parameter definitions, origins, and units are given in Table 29. OUTPUT output parameters are, depending on the value of JK, the appropriate heading, and either the state or co-state entries for each year.

Table 29

OUTPUT INPUT

Parameter	Definition	Units	Origin
JK	State/co-state flag	Dimensionless (integer)	Main program
JJ	Number of entries in the state and co-state	Dimensionless (integer)	Main program
J2	First co-state entry index	Dimensionless (integer)	Main program
J3	Last co-state entry index	Dimensionless (integer)	Main program
MMM	Output flag	Dimensionless (integer)	Main program

Subroutine COSTS(T,M)

After optimal bonuses have been computed and the G-table constructed as a compact representation of yearly modified states (modified by adding appropriate linear multiples of the bonuses), co-state and bonuses, COSTS computes the yearly bonus cost, the penalty cost, and Hamiltonian corresponding to the optimal solution. It also compares total system costs with the costs accruing to the evolution of the uncontrolled systems and performs the net gain analysis.

COSTS input parameter definitions, units, and origins are shown in Table 30. COSTS output parameter definitions and units are given in Table 31.

Table 30

COSTS INPUT

Parameter	Definition	Units	Origin
T	Problem extent	Years (real)	Main program
M	Problem extent	Years (integer)	Main program
N	State/co-state dimension	Dimensionless (integer)	Main program
N1	State dimension	Dimensionless (integer)	Main program
G	Modified state/co-state table	$[\text{Men, men/dollar}]^T$ (real)	GFIX
UU	Bonus array	Dollars (real)	Main program
HP	Supply parameter vector	Men/dollar (real)	PROGO

In starting, COSTS prints out the unmodified G-table. It then calls GFIX, which modifies each yearly entry of the G-table, each such entry consisting of the state/co-state vector. On return from GFIX, COSTS prints out the modified G-table. COSTS then calls STAR for the computation of costs in the uncontrolled system. Next, looping over each year from year zero to year M, COSTS computes the bonus cost as the sum of squares of the products of bonus cost multipliers and bonuses. The penalty cost is computed as a positive semi-definite quadratic form in the difference between current and terminal states. The year's total cost is computed as the sum of penalty and bonus costs. The gain is then computed as the difference between total cost in the uncontrolled and optimally controlled system. The Hamiltonian is computed as the difference between the vector product of the subsequent years state and co-state and the total cost. The Hamiltonian is not computed in the last year for which the subsequent year's state

Table 31

COSTS OUTPUT

Parameter	Definition	Units
TIME	Current year	Year (real)
BCOST	Year's bonus cost	Dollars (real)
PCOST	Year's penalty cost	Dollars (real)
TCOST	Year's total cost	Dollars (real)
GAIN	Year's net gain between optimal system and uncontrolled system cost	Dollars (real)
H	Year's Hamiltonian	Dollars (real)
TOTALS	Cumulative total for penalty cost, bonus cost, total cost, uncontrolled system cost and gain	

and co-state are undefined. Finally, totals for penalty cost, bonus cost, total cost, and net gain are accumulated.

Subroutine STAR(M)

STAR computes the yearly states and associated penalty cost for the uncontrolled system. STAR input parameter definitions, units, and origins are shown in Table 32. STAR's output is the G-STAR-table of yearly uncontrolled manpower levels in each year group in the array of associated finally cost PCSTAR.

STAR's initial action is to pre-set uncontrolled problem dimension. It next sets up the state transition matrix and constant input vector and initial state that characterizes the uncontrolled system. Then it sequentially computes each year's state from the preceding. Having constructed the G-STAR-table of uncontrolled states it then computes the cost quadratic form in the difference between each year's state and the desired terminal state.

Table 32

STAR INPUT

Parameter	Definition	Units	Origin
M	Problem extent	Years (integer)	PROGO
N1	Problem state dimension	Dimensionless (integer)	PROGO
ALPHA	State transition parameters	Dimensionless (real)	PROGO
ELZERO	Initial state	Men (list)	PROGO
ELT	Final state	Men (red)	PROGO

Appendix A

MATHEMATICAL TO PROGRAM SYMBOL MAP

Mathematical Symbol	DBMMP Symbol
L_t	ELVEC
B_t	UU
A	A
β	BB
C	CC
R	RR
S	S
F	F
F_2	F2
F_1	F1
S'	ST
$S'F_2 + B_1F_1$	P1
$(R - S'F_2S - F_1^*)$	W
$(R - S'F_2S - F_1^*)^{-1}$	W
P_1'	PT
A_1	A1
β'	BT
B_1	BA
B_1^{-1}	BA
P	P2
Q	Q2
Y	Y

Mathematical Symbol	DBMMP Symbol
Z	BA
D	B
ϕ	(AR,AI) (complex)
V	R, RINV
μ_i	U
V^{-1}	R
$\text{diag } (\mu_i^T)$	UPØW
$\text{diag } \frac{(1 - \mu_i^T)}{1 - \mu_i}$	CVEC
$V \text{ diag } \mu_i^T V^{-1}$	CTEMP
$\bar{D}$	C
E_{11}	E11
E_{12}	E12
E_{21}	E22
L_0	ELZERO
L_T	ELT
W_1	W
W_2	TERM1
W_3	BTZ
$G_{T,t}$	G
$C_{B,t}$	BCØST
$C_{P,t}$	PCØST
C_{TOT}	TCØST
H_t	H

Mathematical Symbol	DBMMP Symbol
A*	ASTAR
B*	BSTAR
C*	PCSTAR
G_t	GAIN

Appendix B

ECONOMICS MODEL EQUATIONS

The following set of equations is used by subroutine COMPUT of the DBMMP to compute the partials matrix, (F_{ij}). The input parameters $L_{T,i}$, $i = 1, 2, m$, ϕ_1, ϕ_2, μ, ..., are defined and described in Munch (1977).

$$X_1 = \sum_{i=1}^{m/2} L_{T,i} \tag{B.1}$$

$$X_2 = \sum_{i=m/2+1}^{m} L_{T,i} \tag{B.2}$$

$$Q = \left[\delta_1 X_1^{-\rho} + \delta_2 X_2^{-\rho}\right]^{-\frac{\mu}{\rho}} \tag{B.3}$$

$$Z = Q^{\mu} \tag{B.4}$$

$$\rho' = \rho + 1 \tag{B.5}$$

$$q_1 = \delta_1 (Q/X_1)^{\rho'} \tag{B.6}$$

$$q_2 = \delta_2 (Q/X_2)^{\rho'} \tag{B.7}$$

$$w_1 = w_2 q_1 / q_2 \tag{B.8}$$

$$\mu' = \mu - 1 \tag{B.9}$$

$$r = \mu Z/Q \tag{B.10}$$

$$s = X_2 W_2 / (X_1 W_1) \tag{B.11}$$

$$t = Z\varepsilon \tag{B.12}$$

$$z_1 = rq_1 \tag{B.13}$$

$$z_2 = rq_2 \tag{B.14}$$

$$p = W_2/z_2 \tag{B.15}$$

$$z_{11} = z_1 q_1(\mu' - \rho' s)/Q \tag{B.16}$$

$$z_{22} = z_2 q_2(\mu' - \rho'/s)/Q \tag{B.17}$$

$$z_{12} = z_1 q_2(\mu + \rho)/Q \tag{B.18}$$

$$J_{11} = p\left[z_{11} - z_1^2/t\right] \tag{B.19}$$

$$J_{22} = p\left[z_{22} - z_2^2/t\right] \tag{B.20}$$

$$J_{12} = p[z_{12} - z_1 z_2/t] \tag{B.21}$$

$$F_{ij} = \begin{cases} J_{11}, & i = 1, 2, \ldots, m/2, \; j = 1, 2, \ldots, m/2 \\ J_{12}, & i = 1, 2, \ldots, m/2, \; j = \frac{m}{2} + 1, .-., m \;, \\ & i = m/2+1, \ldots, m, \; j = 1, 2, \ldots, m/2 \\ J_{22}, & i = m/2+1, \ldots, m, \; j = m/2+1, \ldots, m \end{cases} \tag{B.22}$$

The economic model and model parameters underlying these equations are given in Munch (1977).